1 Using this book

This book is a do-it-yourself guide and handbook for TEX. Here in this section we tell you how to use the book to maximum advantage.

We recommend that you first either read or skim in sequence Sections 'usebook' through 'examples', which tell you what you need to know in order to get started using TEX. If you've already had experience using TEX, it will still be helpful to know what kinds of information are in these sections of the book. Sections 'concepts'–'tips' and 'capsule', which occupy most of the rest of the book, are designed to be accessed randomly. Nevertheless, if you're the kind of person who likes to read reference manuals, you'll find that it *is* possible to proceed sequentially if you're willing to take a lot of detours at first.

In Section 'usingtex', "Using TEX", we explain how to produce a TEX document from a TEX input file. We also describe the conventions for preparing that input file, explain a little about how TEX works, and tell you about additional resources that are available. Reading this section will help you understand the examples in the next section.

Section 'examples', "Examples", contains a sequence of examples that illustrate the use of TEX. Each example consists of a page of output together with the input that we used to create it. These examples will orient you and help you locate the more detailed material that you'll need as you go. By seeing which commands are used in the input, you'll know where to look for more detailed information on how to achieve the effects shown in the output. The examples can also serve as models for simple documents, although we must caution you that because we've tried to pack a variety of TEX commands into a small number of pages, the examples are not necessarily illustrations of good or complete document design.

As you read the explanation of a command, you may encounter some unfamiliar technical terms. In Section 'concepts', "Concepts", we define and explain these terms. We also discuss other topics that aren't covered elsewhere in the book. The inside back cover of the book contains a list of all the concepts and the pages where they are described. We suggest that you make a copy of this list and keep it nearby so that you'll be able to identify and look up an unfamiliar concept immediately.

TeX's commands are its primary vocabulary, and the largest part of this book is devoted to explaining them. In Sections 'paras' through 'general' we describe the commands. You'll find general information about the command descriptions on page '`cmddesc`'. The command descriptions are arranged functionally, rather like a thesaurus, so if you know what you want to do but you don't know which command does it for you, you can use the table of contents to guide you to the right group of commands. Commands that we think are both particularly useful and easy to understand are indicated with a pointing hand ().

Section 'capsule', "Capsule summary of commands", is a specialized index that complements the more complete descriptions in Sections 'paras'–'general'. It lists TeX's commands alphabetically, with a brief explanation of each command and a reference to the page where it is described more completely. The capsule summary will help you when you just want a quick reminder of what a command does.

TeX is a complex program that occasionally works its will in mysterious ways. In Section 'tips', "Tips and techniques", we provide advice on solving a variety of specific problems that you may encounter from time to time. And if you're stumped by TeX's error messages, you'll find succor in Section 'errors', "Making sense of error messages".

The gray tabs on the side of the book will help you locate parts of the book quickly. They divide the book into the following major parts:

1) general explanations and examples
2) concepts
3) descriptions of commands (five shorter tabs)
4) advice, error messages, and the `eplain.tex` macros
5) capsule summary of commands
6) index

In many places we have provided page references to *The TeXbook* (see page '`resources`' for a citation). These references apply to the seventeenth edition of *The TeXbook*. For other editions, some references may be off by a page or two.

Syntactic conventions

In any book about preparing input for a computer, it's necessary to indicate clearly the literal characters that should be typed and to distinguish those characters from the explanatory text. We use the Computer Modern typewriter font for `literal input like this`, and also for the names of TEX commands. When there's any possibility of confusion, we enclose TEX input in single quotation marks, '`like this`'. However, we occasionally use parentheses when we're indicating single characters such as (`) (you can see why).

For the sake of your eyes we usually just put spaces where you should put spaces. In some places where we need to emphasize the space, however, we use a '␣' character to indicate it. Naturally enough, this character is called a *visible space*.

Descriptions of the commands

Sections 'paras'–'general' contain a description of what nearly every TEX command does. Both the primitive commands and those of plain TEX are covered. The primitive commands are those built into the TEX computer program, while the plain TEX commands are defined in a standard file of auxiliary definitions (see p. '`\plainTeX`'). The only commands we've omitted are those that are used purely locally in the definition of plain TEX (Appendix B of *The TEXbook*). The commands are organized as follows:

- "Commands for composing paragraphs", Section 'paras', deal with characters, words, lines, and entire paragraphs.
- "Commands for composing pages", Section 'pages', deal with pages, their components, and the output routine.
- "Commands for horizontal and vertical modes", Section 'hvmodes', have corresponding or identical forms for both horizontal modes (paragraphs and hboxes) and vertical modes (pages and vboxes). These commands provide boxes, spaces, rules, leaders, and alignments.
- "Commands for composing math formulas", Section 'math', provide capabilities for constructing math formulas.
- "Commands for general operations", Section 'general', provide TEX's programming features and everything else that doesn't fit into any of the other sections.

You should think of these categories as being suggestive rather than rigorous, because the commands don't really fit neatly into these (or any other) categories.

Within each section, the descriptions of the commands are organized by function. When several commands are closely related, they are described as a group; otherwise, each command has its own explanation. The description of each command includes one or more examples and the output produced by each example when examples are appropriate (for some commands they aren't). When you are looking at a subsection containing functionally related commands, be sure to check the end of a subsection for a "see also" item that refers you to related commands that are described elsewhere.

Some commands are closely related to certain concepts. For instance, the `\halign` and `\valign` commands are related to "alignment", the `\def` command is related to "macro", and the `\hbox` and `\vbox` commands are related to "box". In these cases we've usually given a bare-bones description of the commands themselves and explained the underlying ideas in the concept.

The examples associated with the commands have been typeset with `\parindent`, the paragraph indentation, set to zero so that paragraphs are normally unindented. This convention makes the examples easier to read. In those examples where the paragraph indentation is essential, we've set it explicitly to a nonzero value.

The pointing hand in front of a command or a group of commands indicates that we judged this command or group of commands to be particularly useful and easy to understand.

Many commands expect arguments of one kind or another (p. '`arg1`'). The arguments of a command give TEX additional information that it needs in order to carry out the command. Each argument is indicated by an italicized term in angle brackets that indicates what kind of argument it is:

⟨*argument*⟩	a single token or some text enclosed in braces
⟨*charcode*⟩	a character code, i.e., an integer between 0 and 255
⟨*dimen*⟩	a dimension, i.e., a length
⟨*glue*⟩	glue (with optional stretch and shrink)
⟨*number*⟩	an optionally signed integer (whole number)
⟨*register*⟩	a register number between 0 and 255

All of these terms are explained in more detail in Section 'concepts'. In addition, we sometimes use terms such as ⟨*token list*⟩ that are either self-explanatory or explained in the description of the command. Some commands have special formats that require either braces or particular words. These are set in the same bold font that we use for the command headings.

Some commands are parameters (p. '`introparms`') or table entries. This is indicated in the command's listing. You can either use a parameter as an argument or assign a value to it. The same holds for table entries. We use the term "parameter" to refer to entities such as `\pageno` that are actually registers but behave just like parameters.